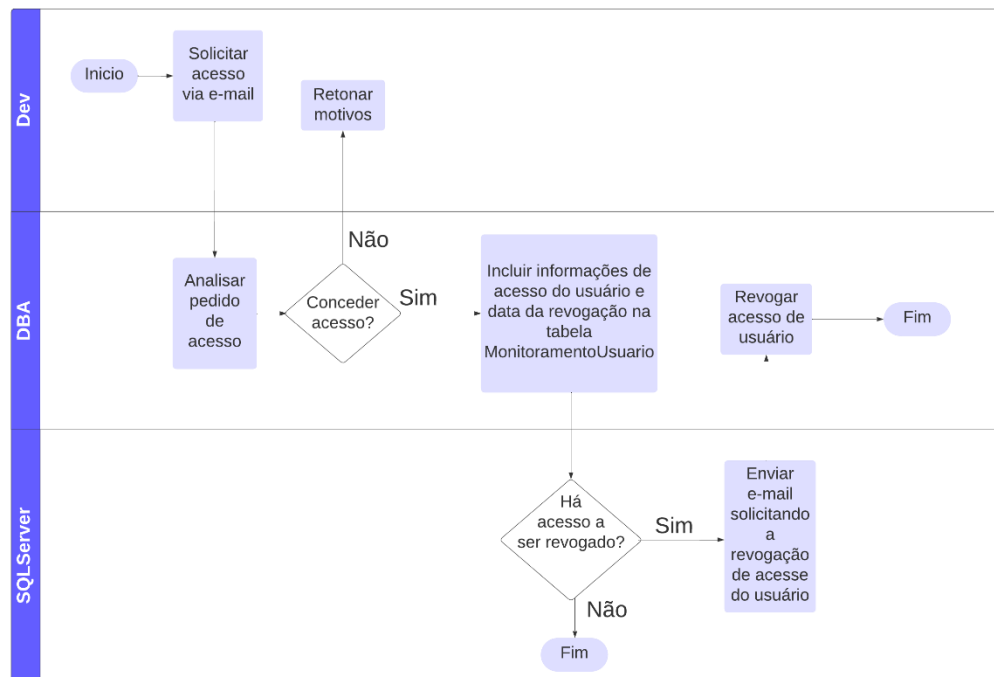


Gerenciamento de acesso as instâncias SQLServer

Sumário

1. Fluxograma.....	3
2. Objetos	3
3. Tabela MonitoramentoUsuario.....	4
4. Procedimento spInserirDadosMonitoramentoUsuario.....	4
5. Job DBA - RevogarAcesso.....	5
6. Procedimento spRevogarPermissao	6
7. Procedimento spMonitoramentoLogin.....	7

1. Fluxograma



2. Ambientes

SRVSQLHOMOLOG, SRVSQLDESENV, SRVSQLHOMOLOG19 e SRVSQLDESENV19

3. Objetos

Banco: DBDataAdm		
Tabelas	Procedimentos	Trabalho
MonitoramentoUsuario	spInserirDadosMonitoramentoUsuario	DBA - RevogarAcesso
	spRevogarPermissao	
	spMonitoramentoLogin	

4. Tabela MonitoramentoUsuario

Tabela: MonitoramentoUsuario	
Tabelas	Descrição
MUsuBancoDeDados	Nome do Banco de dados
MUsuFuncaoBancoDeDados	Função do banco de dados(Permissão) a qual o usuário do banco de dados é vinculado
MUsuUsuario	Usuário do banco de dados
MUsuDataPermissao	Data que a permissão foi concedida ao usuário
MUsuDataRevogarAcesso	Data para revogar permissão do usuário
MUsuDescricao	Campo de observação

5. Procedimento spInserirDadosMonitoramentoUsuario

Procedimento: spInserirDadosMonitoramentoUsuario
Função: Inserir na tabela MonitoramentoUsuario o nome do banco de dados, função de banco de dados(Permissão), usuário, data para revogar acesso e descrição
Sintaxe: <code>exec InserirDadosMonitoramentoUsuario 'NomeBanco','Funcao', 'NomeUsuario', 'Data'.</code> O parâmetro @Descricao é nulo
<pre>USE [Traces] GO SET ANSI_NULLS ON GO SET QUOTED_IDENTIFIER ON GO ALTER PROCEDURE [dbo].[InserirDadosMonitoramentoUsuario] (@BancoDeDados varchar(50), @FuncaoBancoDeDados varchar(50), @Usuario varchar(50), @DataRevogarAcesso nvarchar(50), @Descricao varchar(100) = NULL) AS BEGIN DECLARE @DataPermissao nvarchar(50) = FORMAT(GETDATE(), 'dd/MM/yyyy') -- Inserir os dados na tabela MonitoramentoUsuario INSERT INTO MonitoramentoUsuario (BancoDeDados, FuncaoBancoDeDados, Usuario, DataPermissao, DataRevogarAcesso, Descricao)</pre>

```
VALUES (@BancoDeDados, @FuncaoBancoDeDados, @Usuario,
@DataPermissao, @DataRevogarAcesso, @Descricao)
END
```

6. Job DBA - RevogarAcesso

Job: DBA - RevogarAcesso
<p>Função: Trabalho executa o procedimento spRevogarPermissao diariamente</p> <pre>USE [msdb] GO BEGIN TRANSACTION DECLARE @ReturnCode INT SELECT @ReturnCode = 0 IF NOT EXISTS (SELECT name FROM msdb.dbo.syscategories WHERE name=N'[Uncategorized (Local)]' AND category_class=1) BEGIN EXEC @ReturnCode = msdb.dbo.sp_add_category @class=N'JOB', @type=N'LOCAL', @name=N'[Uncategorized (Local)]' IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback END DECLARE @jobId BINARY(16) EXEC @ReturnCode = msdb.dbo.sp_add_job @job_name=N'DBA - RevogarAcesso', @enabled=1, @notify_level_eventlog=0, @notify_level_email=0, @notify_level_netsend=0, @notify_level_page=0, @delete_level=0, @description=N'Envio de email informando ao DBA quando houver acesso de usuário para ser revogado no banco de dados.', @category_name=N'[Uncategorized (Local)]', @owner_login_name=N'sa', @job_id = @jobId OUTPUT IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback /***** Object: Step [Envio de e-mail] Script Date: 02/02/2024 14:26:50 *****/ EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId, @step_name=N'Envio de e-mail', @step_id=1, @cmdexec_success_code=0, @on_success_action=1, @on_success_step_id=0, @on_fail_action=2, @on_fail_step_id=0, @retry_attempts=0, @retry_interval=0, @os_run_priority=0, @subsystem=N'TSQL', @command=N'EXEC spRevogarPermissao', @database_name=N'DBDataAdm', @flags=0 IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback EXEC @ReturnCode = msdb.dbo.sp_update_job @job_id = @jobId, @start_step_id = 1 IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback EXEC @ReturnCode = msdb.dbo.sp_add_jobschedule @job_id=@jobId, @name=N'Diarimamente',</pre>

```

        @enabled=1,
        @freq_type=4,
        @freq_interval=1,
        @freq_subday_type=1,
        @freq_subday_interval=0,
        @freq_relative_interval=0,
        @freq_recurrence_factor=0,
        @active_start_date=20240202,
        @active_end_date=99991231,
        @active_start_time=80000,
        @active_end_time=235959,
        @schedule_uid=N'05faa521-4b50-43cf-bea2-b0f96aea979d'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_add_jobserver @job_id = @jobId, @server_name =
N'(local)'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
COMMIT TRANSACTION
GOTO EndSave
QuitWithRollback:
    IF (@@TRANCOUNT > 0) ROLLBACK TRANSACTION
EndSave:
GO

```

7. Procedimento spRevogarPermissao

Procedimento: spRevogarPermissao

Função: Enviar e-mail informando ao DBA quando houver acesso de usuário para ser revogado no banco de dados.

```

USE [DBDataAdm]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE spRevogarPermissao
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM MonitoramentoUsuario
        WHERE CONVERT(DATE, MUsuDataRevogarAcesso, 103) = CONVERT(DATE,
GETDATE(), 103)
    )
    BEGIN
        EXEC msdb.dbo.sp_send_dbmail
            @profile_name = 'Data Adm',
            @recipients = 'caroline.mattos@ctctech.com.br',
            @subject = 'Revogar permissão - SRVSQLHOMOLOG19',
            @body = 'Há permissão em
SRVSQLHOMOLOG19.DBDataAdm.dbo.MonitoramentoUsuario para ser revogada na data de
hoje',
            @body_format = 'TEXT';
    END
END

```

8. Procedimento spMonitoramentoLogin

Procedimento: spMonitoramentoLogin
Função: Retornar banco de dados, login, usuário, função de banco de dados e status do login na instancia, onde login não é nulo.
Sintaxe: <code>exec spMonitoramentoLogin</code> ou <code>exec spMonitoramentoLogin 'Login'</code> A primeira opção exibe as informações de acesso de todos os usuários que possuem vínculo aos logins da instancia. Enquanto a segunda filtra o resultado através do login declarado.
<pre>USE [Traces] GO SET ANSI_NULLS ON GO SET QUOTED_IDENTIFIER ON GO ALTER PROCEDURE [dbo].[MonitoramentoLogin] @UserLogin NVARCHAR(100) = NULL AS BEGIN IF EXISTS (SELECT * FROM TEMPDB.dbo.sysobjects WHERE NAME IN ('##Users')) BEGIN DROP TABLE ##Users END IF EXISTS (SELECT * FROM TEMPDB.dbo.sysobjects WHERE NAME IN (N'##ACESSO')) BEGIN DROP TABLE ##ACESSO END CREATE TABLE ##Users ([sid] VARBINARY(100) NULL, [Login Name] VARCHAR(100) NULL) CREATE TABLE ##ACESSO ([uSER ID] VARCHAR(MAX), [sERVER LOGIN] VARCHAR(MAX), [DATABASE ROLE] VARCHAR(MAX), [DATABASE] VARCHAR(MAX), [LoginDisabled] BIT) DECLARE @DBName NVARCHAR(255) DECLARE db_cursor CURSOR FOR SELECT name FROM sys.databases --WHERE database_id > 4 OPEN db_cursor FETCH NEXT FROM db_cursor INTO @DBName WHILE @@FETCH_STATUS = 0</pre>

```

BEGIN
    DECLARE @cmd1 NVARCHAR(MAX)
    SET @cmd1 = '
        INSERT INTO ##Users ([sid],[Login Name]) SELECT sid, loginname FROM
master.dbo.syslogins;

        INSERT INTO ##ACESSO
        SELECT
            su.[name],
            u.[Login Name],
            sug.name,
            @DatabaseName,
            CASE WHEN sp.is_disabled = 1 THEN 1 ELSE 0 END -- Adicionando status de
login desativado
        FROM
            ' + QUOTENAME(@DBName) + '.dbo.sysusers su
        LEFT OUTER JOIN
            ##Users u ON su.sid = u.sid
        LEFT OUTER JOIN
            (' + QUOTENAME(@DBName) + '.dbo.sysmembers sm
            INNER JOIN ' + QUOTENAME(@DBName) + '.dbo.sysusers sug ON
sm.groupuid = sug.uid) ON su.uid = sm.memberuid
        LEFT OUTER JOIN
            master.sys.server_principals sp ON u.[Login Name] = sp.name
        WHERE
            su.hasdbaccess = 1
            AND (@UserLogin IS NULL OR (su.[name] != "dbo" AND u.[Login Name] =
@UserLogin));
    '

    EXEC sp_executesql @cmd1, N'@UserLogin NVARCHAR(100), @DatabaseName
NVARCHAR(255)', @UserLogin = @UserLogin, @DatabaseName = @DBName

    FETCH NEXT FROM db_cursor INTO @DBName
END

CLOSE db_cursor
DEALLOCATE db_cursor

SELECT
    [uSER ID] AS Usuario,
    [sERVER LOGIN] AS Login,
    [DATABASE ROLE] AS FuncaoBancoDeDados,
    [DATABASE] AS BancoDeDados,
    [LoginDisabled] AS LoginDesabilitado
FROM ##ACESSO
WHERE [sSERVER LOGIN] IS NOT NULL
GROUP BY
    [uSER ID], [sSERVER LOGIN], [DATABASE ROLE], [DATABASE], [LoginDisabled]
END

```